BIDO User Manual Version 1.2

BIDO is a package of analysis codes to identify properties of surface waves using circular-array records of microtremors (ambient vibrations; bidô in Japanese).

Ikuo Cho and Taku Tada

Original text in Japanese written by I. Cho and translated into English by T. Tada

Contact: Ikuo-chou@aist.go.jp

http://staff.aist.go.jp/ikuo-chou/bidodl_en.html

October 7, 2009

Contents

1. Outline……3

A1.1 Array Exploration of Microtremors
A1.2 The SPAC Method

2. For Whom It Is Meant …… 4

3. Usage …… 4

4. Technical Information / How to Install …… 4

5. Program Description and Algorithm …… 5

6. Example of Program Execution Using Demo Data …… 8

A6.1 Details of the Synthetic Data
A6.2 The Dialogue that Appears on Activating the Program
A6.3 The seismfile
A6.4 The segment File
A6.5 Analysis Results (folder RESULT)
A6.6 Analysis Results (folder RESULT/ave)
A6.7 Analysis Results (folders with alphanumeric names)
A6.8 One Approach to Make the Most of the Method's Potential
A6.9 A Knack for Setting Parameters

7. Warnings / Download…… 31

A7 Citation

Appendix Execution log

1. Outline

BIDO is an analysis tool we offer free for the array exploration of microtremors**.** The software can be used to identify properties of surface waves that travel on the ground surface by analyzing circular-array records of microtremors.

Since 2000, our research group, centered on the (former) Shinozaki laboratory at the Tokyo University of Science (joint research by Professor Yuzo Shinozaki, Dr Taku Tada, myself [Ikuo Cho] and graduate students) have undertaken generalization of the SPAC method theory (Reference [2]), and have developed methods that allowed phase velocities of Rayleigh waves to be identified into much longer wavelength ranges than the traditional SPAC method (the CCA method [1,3] and other derivative methods [4]). Our theories have also made it possible to identify phase velocities of Love waves [5], signal-to-noise ratios [3], horizontal-to-vertical amplitude ratios (R/V spectra) [2] of Rayleigh waves, and Rayleigh-to-Love power ratios [5] with simple methods unknown in traditional approaches. BIDO is an analysis tool for microtremor data (circular-array records) that uses these methods to identify properties of surface waves.

A1.1 Array Exploration of Microtremors

The ground surface is constantly trembling because of industrial activities, ocean waves and winds. They are, of course, too small to be felt by human bodies, and can be detected only by microtremor sensors (high-sensitivity seismic sensors). These small tremors are called microtremors (random noise). Simultaneous measurement using more than one microtremor sensors installed on the ground surface is called array measurement of microtremors. Array measurement of microtremors allows one to infer phase velocities of surface waves (propagation velocities of waves called Rayleigh waves and Love waves), on the basis of which one can then infer soil properties (velocity structures). By the term "array exploration of microtremors" we refer to the whole flow of procedures that start with array measurement of microtremors and end with evaluation of subsurface structures.

A1.2 The SPAC Method

A technique of microtremor exploration to analyze phase velocities.

There are two major categories of analysis methods for phase velocities, namely the Capon method (also known as the FK method) and the spatial autocorrelation (SPAC) method. The spatial autocorrelation method was published by Keiiti Aki in 1957, whereas the Capon method was published by Jack Capon in 1969. The spatial autocorrelation method could be called more classic in that sense, but one had to await the activities of Hokkaido University's Hiroshi Okada and coworkers (publication years: 1983-200?) and the University of Tokyo's Kazuyoshi Kudo and coworkers before one could see it becoming a practical method of microtremor exploration. It is only after they began their activities that the spatial autocorrelation method came to be known by a diminutive(?) acronym, the SPAC method.

The SPAC method is strictly constrained by the requirement that the seismic array should be circular (a disadvantage), but it is also characterized by the ability to analyze wavelengths that are fairly long relative to the array size (an advantage). The number of seismic sensors in the circular array can be relatively small, and may even be reduced to

just one central sensor plus one peripheral sensor (!?) in an ultimate case, according to an idea that emerged recently. We have also endeavored to help build a theoretical framework for this idea, which bore fruit in a recent publication (Cho et al., 2008).

An overview of international publications suggests that the SPAC method began to obtain worldwide recognition in the mid-1990s.

Cho, I., T. Tada, and Y. Shinozaki, 2008, Assessing the applicability of the spatial autocorrelation method: A theoretical approach, J. Geophys. Res., 113, B06307, doi:10.1029/2007JB005245.

2. For Whom It Is Meant

We have meant this code package to be used by engineers with expertise in microtremor exploration and by researchers investigating microtremor exploration and array analysis methods (the program is easy to use, but you need expertise to interpret the output appropriately and to infer soil properties).

3. Usage

1) Create microtremor waveform data files, one for each sensor, and place them all within a single folder.

2) On activating the program, you will be asked for a number of parameters. With appropriate input given, the program automatically proceeds and plots the analysis results for you. The results are stored in an output folder that is automatically created under the data folder.

See Example of Program Execution Using Demo Data (Section 6) for details.

4. Technical Information / How to Install

Basic Information

A string of core programs written in Fortran (compilers: g77 and ifort; unchecked for others), linked together via B shell, are executed one after another. Operability confirmed on Windows (XP and Vista) and Linux (Fedora 10).

If you are using Windows, simply download the archive, decompress and execute. The registry is not rewritten, so simply dump the folder into the recycle bin to uninstall. The folder may be saved anywhere, like on the Desktop or in the root of the C drive.

If you are using Linux, it is recommendable to recompile the Fortran codes, although executable files are included in the package. To compile and install, decompress the archive, enter the src directory, and execute Install_linux.sh. Gnuplot (free) is used in drawing graphs, so you have to install it separately unless it is already installed on your PC.

Program Implementation on Windows

The development and operation environment is based on Linux. For use on Windows, the Fortran source codes were compiled using Cygwin (free), and a shell environment was implemented using MSYS (free). The graphic tool gnuplot is also included in the package (compilation finished, but bundled with source codes). If you are to rewrite source codes, you have to recompile them after installing Cygwin on your PC. To compile and install, decompress the archive, enter the src directory, and execute Install_win.sh.

PC Performance Requirement (Example)

The following is an example of the PC performance requirement, described for the case of 6. Demo Data processing under this program's development environment. On a Windows XP/Linux dual boot PC (CPU: Intel(R) Pentium(R) D CPU 3.00 GHz; memory: 2 GB), the CPU time requirement ("user" output of the "time" command) was about 2 min 30 sec on both operation systems (g77 compiler used in both), and the memory requirement was about 300 MB (adjustable by modifying array dimensions by editing PARAM.h when compiling source codes). However, the real time requirement ("real" output of the "time" command) was a little short of 3 min on Linux, while it was about 15 min on XP (shell processing by MSYS and file I/O may possibly account for the time on XP). If this difference can be generalized to all cases is difficult to say, but this outcome seems to recommend the use of Windows only for trial runs and Linux for massive calculations (with the Intel compiler=ifort).

The program itself only occupies about 22 MB of hard disk space (both on Windows and Linux), but this demo requires nearly 120 MB (including the program itself). If you set parameters so that all intermediary data are deleted except for the final analysis results and minimal log files (you will be asked about the choice on activating the program), only less than 50 MB worth of files will be left when the calculations are over (including the program itself).

5. Program Description and Algorithm

● Program Description

\BIDO-win.bat A batch file to activate MSYS on Windows. Not used on Linux.
\bin Executable files for use on Windows are stored in \bin\winbin, while executable files for use on Linux are stored in \bin\linbin. Executable files for use on MSYS are stored just beneath \bin.
\demo Contains demo data (used both Windows and Linux).
\etc Contains scripts to activate MSYS. Not used on Linux.
\run.sh A B shell script to activate the program (used both Windows and Linux).
\script Contains B shell scripts to link Fortran codes (used both Windows and Linux).
\src Contains Fortran codes (used both on Windows and Linux).

● Algorithm

Description of the general flow and individual procedures. In parentheses are the file names of relevant B shell scripts and Fortran codes.

General Flow (\script\circle.sh)

1) Select portions of the data that are good to use.
2) Azimuthally average data around the circle (output from step 1) is not used here)
3) Estimate spectral densities by using the output from steps 1) and 2).
4) Estimate spectral ratios, phase velocities, NS ratios etc.
5) Repeat steps 3) and 4) as many times as there are segment clusters.
6) Calculate means and standard deviations using the output from step 4).
7) Plot the output from step 6).

[1] Automatic Selection of Segments (\script\mksegment.sh, \src\evalrms.F, \src\segment.F)

Segments are selected as follows:

1) The following procedure is performed on all components of all sensors.
   - Subtract a linear trend from the original waveform data and calculate the RMS (let this be called RMS_{all}).
   - For every portion of the data with a prescribed segment duration into which the original waveform is divided (the portions are extracted so that they mutually overlap by half), subtract a linear trend and calculate the RMS. Normalize the RMS values by RMS_{all} (let these be called RMS_{seg}).
2) Make a histogram of all RMS_{seg}, for all components of all sensors, at intervals of 0.1. Identify the interval of the largest frequency in the histogram.
3) Pick up data segments in which all RMS_{seg} of all components and all sensors fall into the interval of the largest frequency simultaneously. Mark them as the segments good to use, and catalog them in the segment file.

[2] Azimuthal Averaging of Data around the Circle (\script\mkcrcle.sh, \src\mkcrcl_*.F)*

All methods adopted in this program start with taking weighted azimuthal averages of records around the circumference. Weighted azimuthal averaging corresponds to calculating Fourier coefficients in the Fourier series expansion around the circle. Our program calculates Fourier coefficients of the zeroth and first orders by default. Our theory is adaptable to unevenly spaced sensors around the circle, and adaptability to practical cases has been investigated for the CCA method (Reference [1]). We have not, however, closely investigated the adaptability for all methods, so we recommend the use of equidistant arrays to the extent that that is possible (we are particularly uncertain about the adaptability to methods that use cross-spectral densities).

[3] Estimating Spectral Densities (\script\estspec.sh, \src\estspec.F)*

This program estimates spectral densities by using both techniques of segment averaging and spectral smoothing for the raw, FFT'ed spectral densities (Bendat & Piersol, 1971). The raw spectral densities are smoothed with a Parzen window before they are segment-averaged. The segment duration and the number of segments in the averaging over multiple segments are given in the dialogue on activating the program: "Duration of data segments for the evaluation of spectra" and "Number of data segments

over which averages are taken. Enter 0 or a very large number if you wish to use all segments simultaneously." The bandwidth of smoothing with a Parzen window is given in the dialogue: "Band width of the Parzen spectral window."

[4] Calculating Spectral Ratios (\script\specratio.sh)*

Ratios are taken, with no frills, between spectral densities estimated in the above-described procedure, except when the denominator is zero. Different types of spectral ratios are linked to the phase velocities via formulae described in References [2, 5]**.** Note that the autocorrelation coefficient of the SPAC method is defined here by a spectral ratio according to the formulation of Reference [2] (in usual practice, the SPAC coefficient is calculated as an azimuthal average of complex coherences between the central point and a peripheral point).

Calculating the H/V spectrum

Starting with Version 1.2.2, I added a feature that estimates horizontal-to-vertical (H/V) spectral ratios, provided that the data have three components, at the one station that is indicated at the top of the seism file (A6.3) (the power of horizontal motion is defined as the sum of the NS and EW component powers). Accordingly, even when the seism file (A6.3) describes a single measurement station alone (even if this does not constitute an array), H/V spectral ratios are calculated as long as there are three-component records. Once the calculation is over, the logarithmic mean and standard deviation are plotted.

Calculating Phase Velocities (\script\spec2pv.sh, \src\sctr2pv.F)*

Spectral ratios are equated to Bessel functions according to the formulae, and a root-solving method that combines bisection and the secant method (Shampine & Watts, 1970) is used to search for *rk* (radius times wavenumber), the argument of the Bessel functions, in the range [0, $rk_{\max}$]. $rk_{\max}$ corresponds to the first maximum or minimum of the function value. The *rk* obtained is used to calculate the phase velocity, $c=2\pi f/k$ (*f* stands for frequency).

[5] Repeat Steps [3] and [4] as Many Times as There Are Segment Clusters

[6] Calculating Means and Standard Deviations (\script\mkave.sh, \src\calave.F)

If the number of segments (number of segments over which averages are taken when estimating spectral densities with the segment averaging method. In other words, the integer value that you enter in the dialogue "Number of data segments over which averages are taken. Enter 0 or a very large number if you wish to use all segments simultaneously" on activating the program) satisfies

(number of all segments catalogued in the segment file(A6.4)) > (number of segments),

then more than one spectral density estimates are obtained from the given waveform data. If we define

(number of segment clusters) = (number of all segments catalogued in the segment file(A6.4)) / (number of segments),

there will be as many estimates of spectral densities, spectral ratios derived from them and phase velocities as the number of segment clusters (the remainder of division is discarded). This program calculates means and standard deviations on the basis of those estimates.

When the program is executed, folders with names \RESULT\(a number) are generated beneath the data folder. The number here represents that of a segment cluster, and these numerically named folders contain the corresponding analysis results. The \RESULT\ave folder contains output of statistical processing of the analysis results stored in those numerically named folders.

To eliminate "outliers," the maximum and minimum values are excluded from the statistical processing if and only if there are more segment clusters than the number NROBUST4AVERAGE_INC indicated in \src\PARAM.h. NROBUST4AVERAGE_INC is set at 8 by default. Some of the analysis results are averaged arithmetically and others logarithmically. Logarithmic averaging is used when averaging ratios like AmpRV_R.d, nsr.d, nsrlim_cca.d, nsrlim_cca.lwapx.d and powratio_R2L.d **(**A6.6).

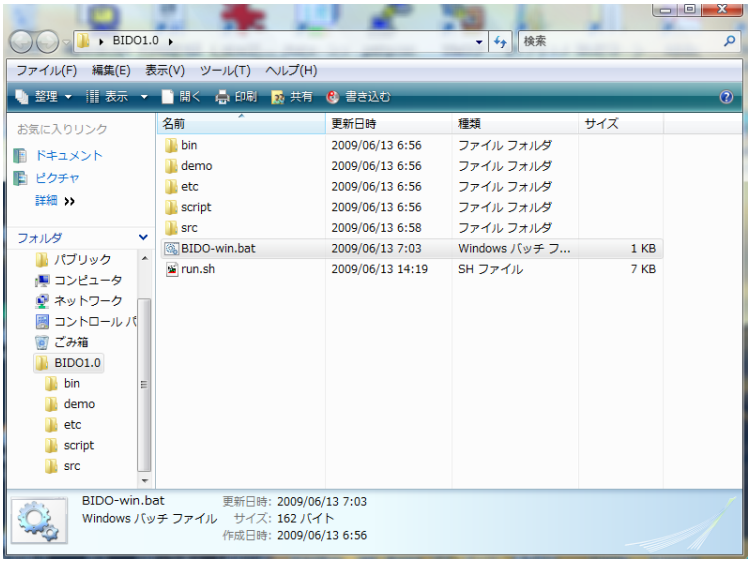
Bendat, J. S., and A. G. Piersol, *Random Data: Analysis and Measurement Procedures*, John Wiley & Sons, 1971.

Shampine, L. F., and H. A. Watts, FZERO, a root-solving code, Report SC-TM-70-631, Sandia Laboratories, 1970.

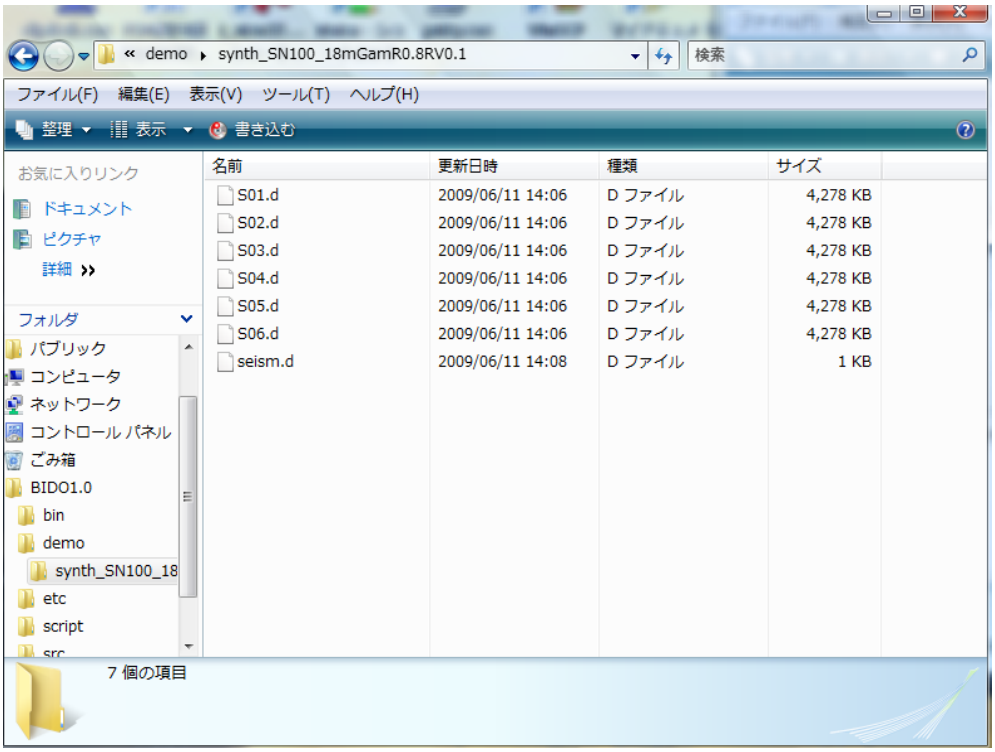
6. Example of Program Execution Using Demo Data

The following is an illustration of demo execution on Windows (the flow is basically the same on Linux).

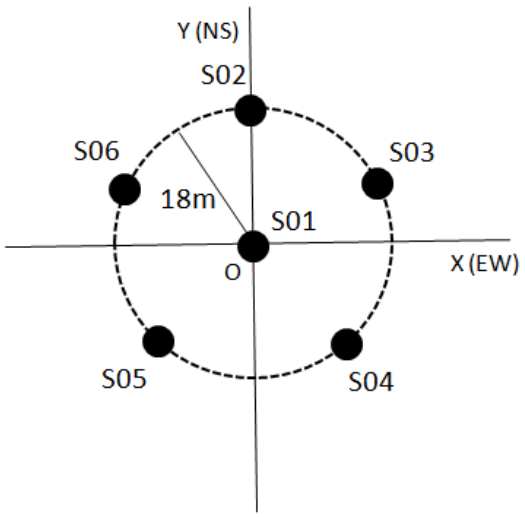
Decompress BIDO1.2.tgz. You will find BIDO1.2 contains the following files:



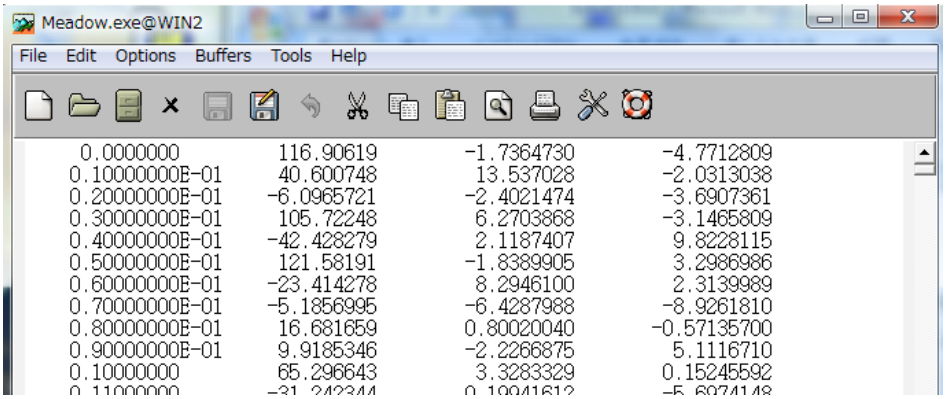
The folder demo\synth_SN100_18mGamR0.8RV0.1 contains synthetic data for demo analysis.



The six files named S0x.d are microtremor waveform data obtained by setting six seismic sensors, each named S01 to S06, in the configuration shown below. These are artificial data, synthesized numerically for the sake of demonstration. All three components were synthesized with a postulated sampling time interval of 0.01 sec and a duration of 10 min. See A6.1 for details on the microtremor waveform synthesis.



Open S01.d with any editor, and you will find:



The leftmost column shows time, whereas the rest stand for the amplitudes of the vertical (z), EW (x) and NS (y) components, from left to right.

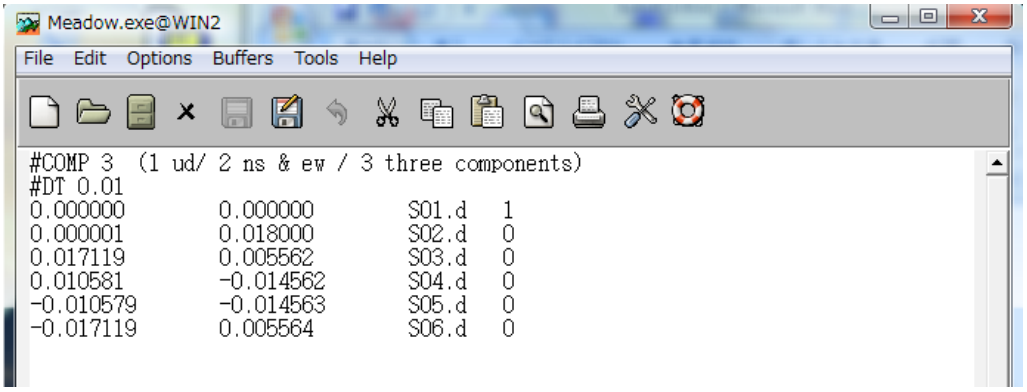
When you analyze your own measurement data, format them in four columns if they have all three components. If there are two horizontal components alone, format them in the same way by inserting zeroes or any dummy data in place of the vertical component. If you have the vertical component alone, let there be just two columns for time and the vertical component (there can be four columns as in the case of three-component data, but there is no need for inserting dummy data). This comes about because my measurements were often for the vertical component alone but were never for the horizontal components alone (the seismometer setup dictated that measurement of the horizontal components was always accompanied by measurement of the vertical component). If you find this inconvenient to handle, modify the Fortran codes for reading data, \src\mkcrcl_uneven.F and \src\mkcrcl_center.F, recompile them and reinstall. Search for lines that say "read(cline" in these two programs (there are three of them in each), and change the order of variables to be read for the cases of Ncomp=1, 2 and 3.

The analysis method of this program presupposes that waveforms are sampled at equal time intervals. The time column should therefore be unnecessary as long as the start time and the sampling time interval are given (you can even do without the start time in practical array data processing). If you are already familiar with array analysis of microtremors, you may wonder why the first column is necessary. In fact, you will find out, on closer look into the above-mentioned code for reading data, that the string of time data in the first column are dummy and are not read (whatever figures you may put in the first column have no influence on the analysis results). I created this column simply because the graphics tool, which I used to check the measurement data, required a format of time and amplitude pairs (directly analyzing waveforms, just plotted and checked, was the most efficient way). As I will mention below, the sampling time interval is given in a file named seism.d.

Anyway, as long as the data are aligned in this way, the data type can be velocity, acceleration or anything, and can be of any unit. There is no constraint on the format of the values (with an exponent part or with a floating point). Separators between values

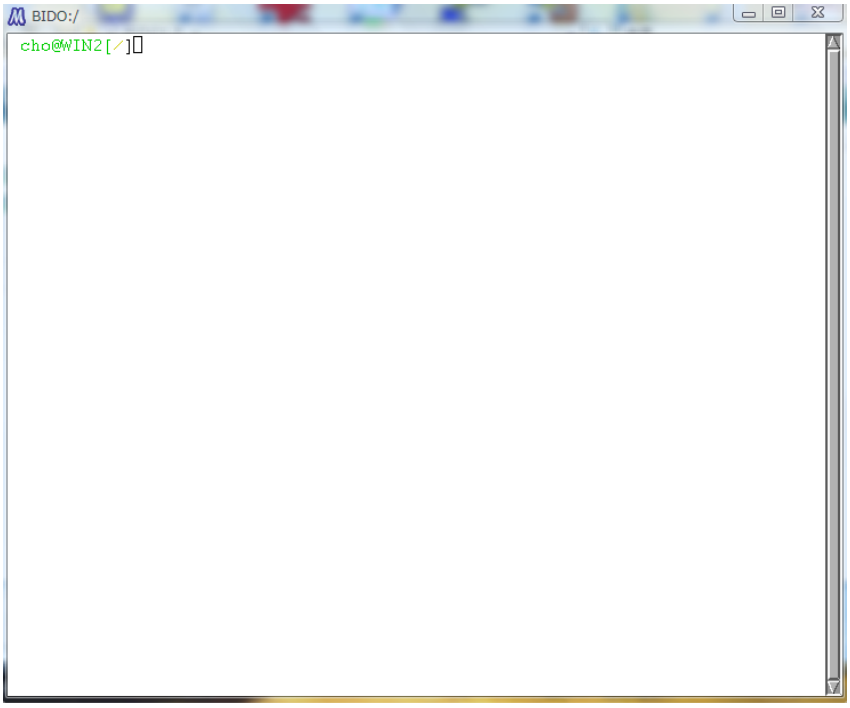
can either be spaces, tabs or commas. There is no particular rule on the naming of data files (they can even lack the extension *.d). When you have created data files, one for each seismic sensor, place them all within a single folder. Let all waveforms in the data files start at time zero.

You will find a file named seism.d in the same folder. Open it to find:

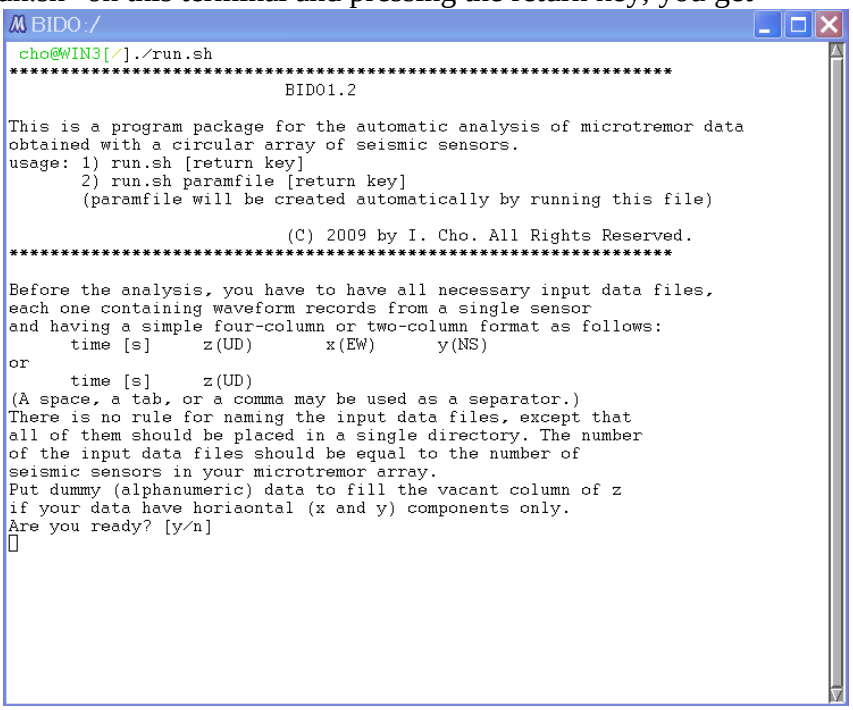


The figure in the first row indicates the components to be used in the analysis. In the second row is the sampling time interval of the waveform records (#COMP and #DT are a sort of spells and should not be omitted). In the third row and below are x and y coordinates (km), data file names and center/periphery IDs (1 for center and 0 for periphery). The file name seism.d should not be modified, and it should be placed within the same folder as the measurement data. This file need not, however, necessarily have been created beforehand. When this file is not found in the data folder, the program simply asks for necessary information and automatically creates one.

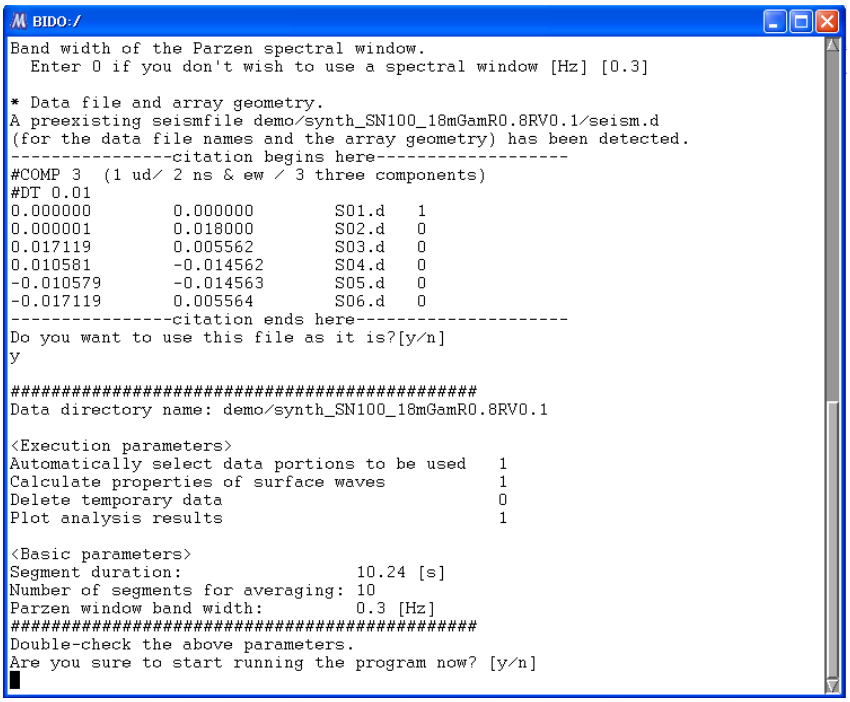
Now that you know what data set there is, let us start the analysis. Go back to just beneath BIDO1.2 and double-click BIDO-win.bat. This opens a window (terminal) as follows (this action is not necessary on Linux 3):



Typing "run.sh" on this terminal and pressing the return key, you get

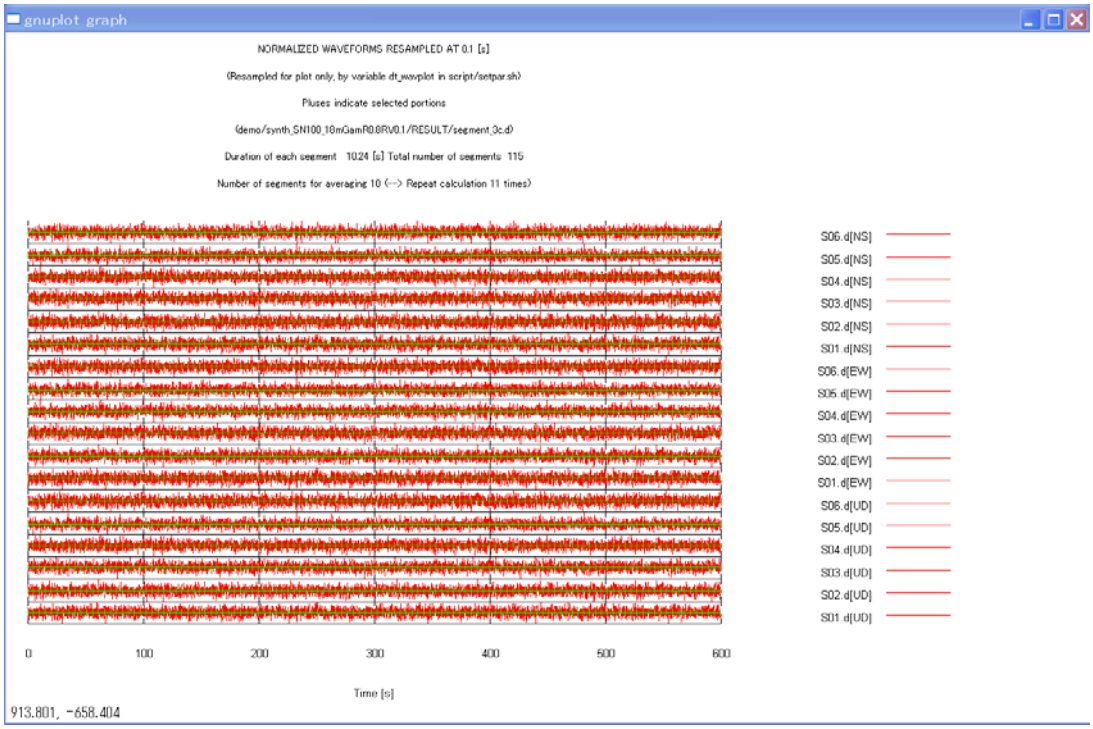

BIDO:/
cho@WIN3[/]./run.sh
****
BIDO1.2
This is a program package for the automatic analysis of microtremor data
obtained with a circular array of seismic sensors.
usage: 1) run.sh [return key]
2) run.sh paramfile [return key]
(paramfile will be created automatically by running this file)
(C) 2009 by I. Cho. All Rights Reserved.
****
Before the analysis, you have to have all necessary input data files,
each one containing waveform records from a single sensor
and having a simple four-column or two-column format as follows:
time [s] z(UD) x(EW) y(NS)
or
time [s] z(UD)
(A space, a tab, or a comma may be used as a separator.)
There is no rule for naming the input data files, except that
all of them should be placed in a single directory. The number
of the input data files should be equal to the number of
seismic sensors in your microtremor array.
Put dummy (alphanumeric) data to fill the vacant column of z
if your data have horiaontal (x and y) components only.
Are you ready? [y/n]

In the above and all following screens, simply type "y" or press the return key (to use default parameters). You will finally come down to the following screen:


BIDO:/
Band width of the Parzen spectral window.
Enter 0 if you don't wish to use a spectral window [Hz] [0.3]
* Data file and array geometry.
A preexisting seismfile demo/synth_SN100_18mGamR0.8RV0.1/seism.d
(for the data file names and the array geometry) has been detected.
----------citation begins here----------
#COMP 3 (1 ud/ 2 ns & ew / 3 three components)
#DT 0.01
0.000000 0.000000 S01.d 1
0.000001 0.018000 S02.d 0
0.017119 0.005562 S03.d 0
0.010581 -0.014562 S04.d 0
-0.010579 -0.014563 S05.d 0
-0.017119 0.005564 S06.d 0
----------citation ends here----------
Do you want to use this file as it is?[y/n]
y
##########
Data directory name: demo/synth_SN100_18mGamR0.8RV0.1
<Execution parameters>
Automatically select data portions to be used 1
Calculate properties of surface waves 1
Delete temporary data 0
Plot analysis results 1
<Basic parameters>
Segment duration: 10.24 [s]
Number of segments for averaging: 10
Parzen window band width: 0.3 [Hz]
##########
Double-check the above parameters.
Are you sure to start running the program now? [y/n]

You will find the parameters you entered between two rows of "###..." If you are to analyze your own measurement data, be sure to check out detailed descriptions of the questions asked (A6.2) and a knack for setting parameters (A.6.8). You are now being asked if it is all right to start the analysis with the values presented, so typing "y" will launch the analysis, which will proceed automatically.

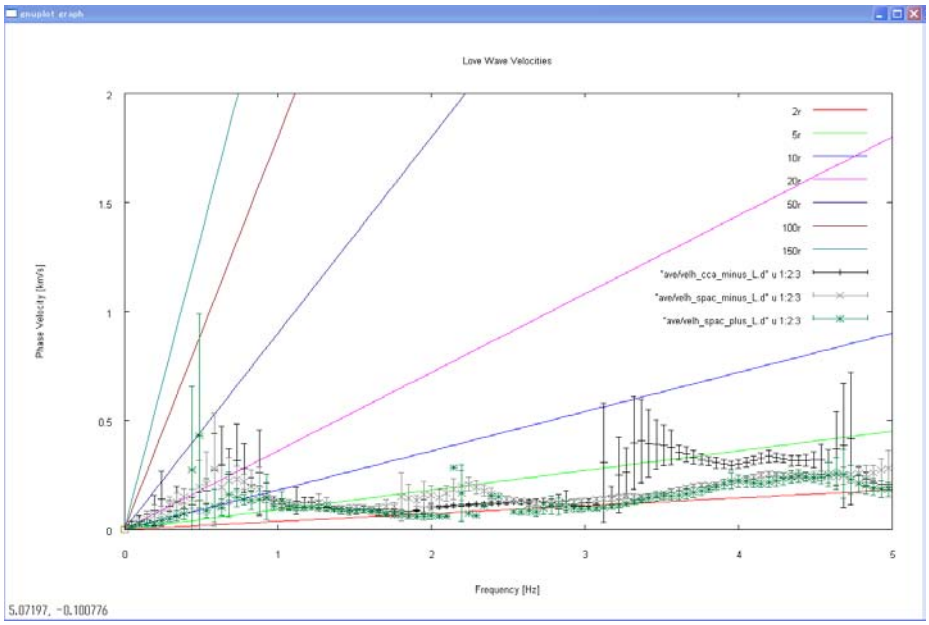
Soon after analysis begins, prior to the data processing to estimate the spectra, there appears a plot of the waveform records, and information on which parts of them will be used in the spectral analysis (see below. Waveforms, and the parts of them used in the spectral analysis, are illustrated with red and green plus signs, respectively).



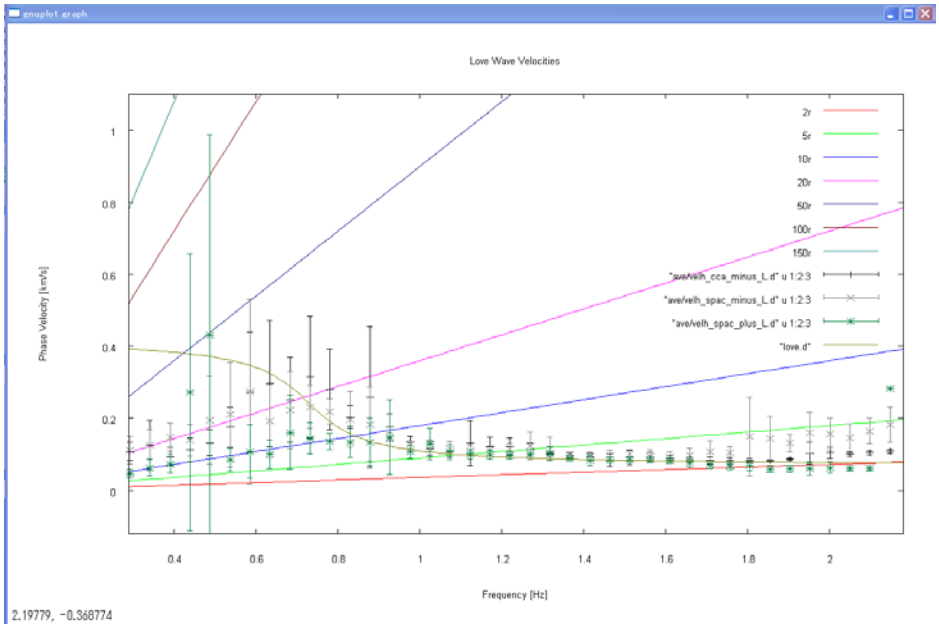
Going back to the interactive terminal screen and pressing the return key erases this plot and kicks off the spectral analysis.
The execution log of the analysis looks like this (Appendix).
This log is automatically generated in the folder where the data files are stored.

When the analysis is over, (1) phase velocities of Love waves, (2) phase velocities of Rayleigh waves, (3) an R/V spectrum of Rayleigh waves, (4) the power shares of Rayleigh waves in horizontal motion, (5) the H/V spectrum, (6) comparison of the R/V spectrum of Rayleigh waves and the H/V spectrum, and (7) NS ratios of horizontal motion are displayed in plots.

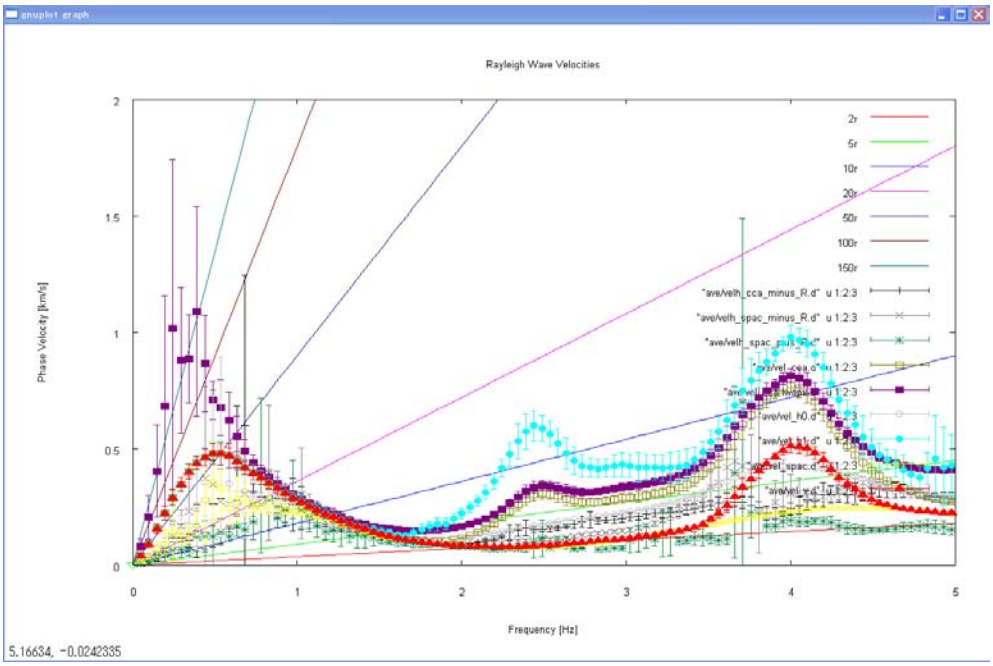
1) Phase velocities of Love waves are shown in data points with error bars (standard deviations). The horizontal axis, frequency; the vertical axis, phase velocity.



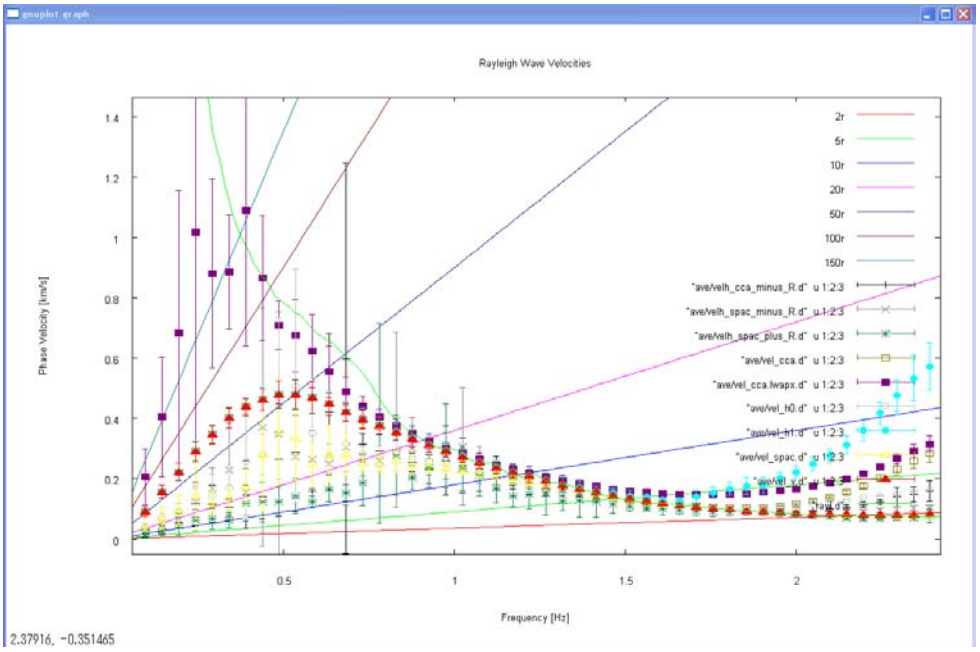
Analysis results from three methods, called the CCA-L, SPAC-L and SPAC+L methods, are shown simultaneously. The straight lines radiating from the origin stand for wavelengths of 2, 5, etc. times the array radius *r* (just for reference). The prescribed Love-wave phase velocities (A6.1) seem to be reproduced in a band of approximately 1-2 Hz. For reference, below is an enlarged view of the comparison, approximately between 0.4 and 2.2 Hz, between the phase velocities prescribed in creating the synthetic data (solid curve) and the analysis results. With gnuplot, it is fairly easy to enlarge certain parts of a plot. See gnuplot manual pages for details.



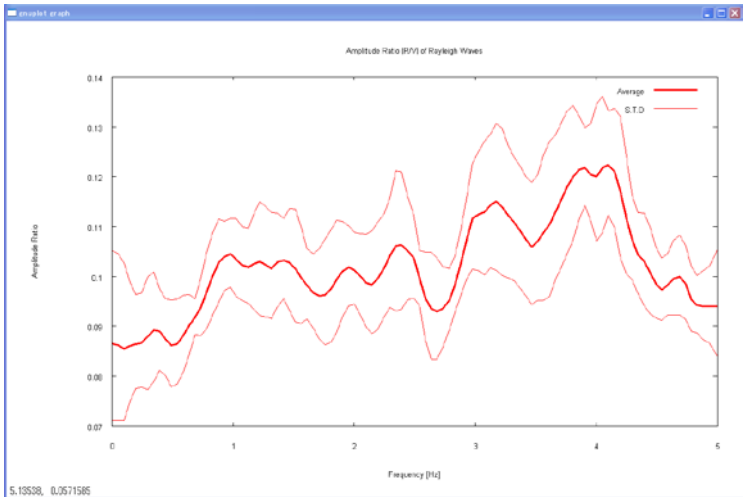
2) Go back to the terminal and press the return key. This will take you to the following plot of the analysis results for phase velocities of Rayleigh waves.



Analysis results from the CCA-R, SPAC-R and SPAC+R methods (using horizontal-motion records) and the CCA, nc-CCA, H0, H1 and V methods (using vertical-motion records) are plotted simultaneously. The Rayleigh-wave phase velocities prescribed (A6.1) seem to be reproduced in a band of approximately 0.5-2 Hz. For reference, below is an enlarged view of the comparison, up to about 2 Hz, between the phase velocities prescribed in creating the synthetic data (solid curve) and the analysis results. The nc-CCA method seems to retain analysis capabilities down to the lowest frequency.

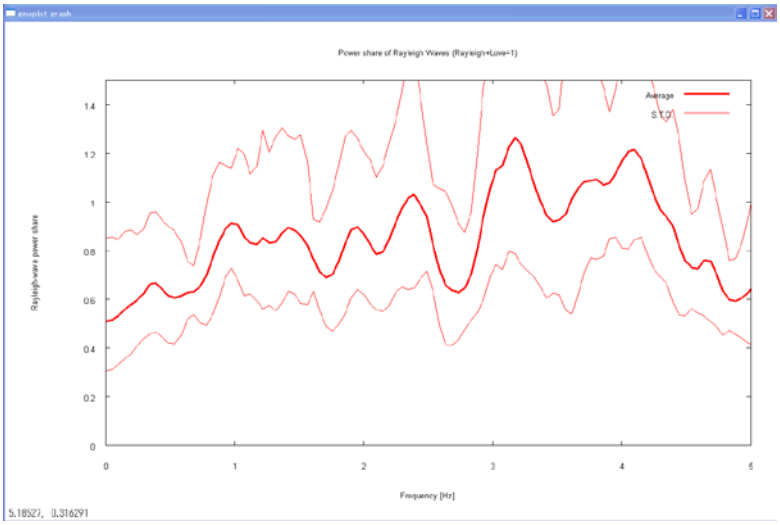


3) Go back to the terminal and press the return key. This plots the R/V spectrum (amplitude ratio between the horizontal motion and the vertical motion) of Rayleigh waves (heavy curve, mean; light curves, standard deviation).



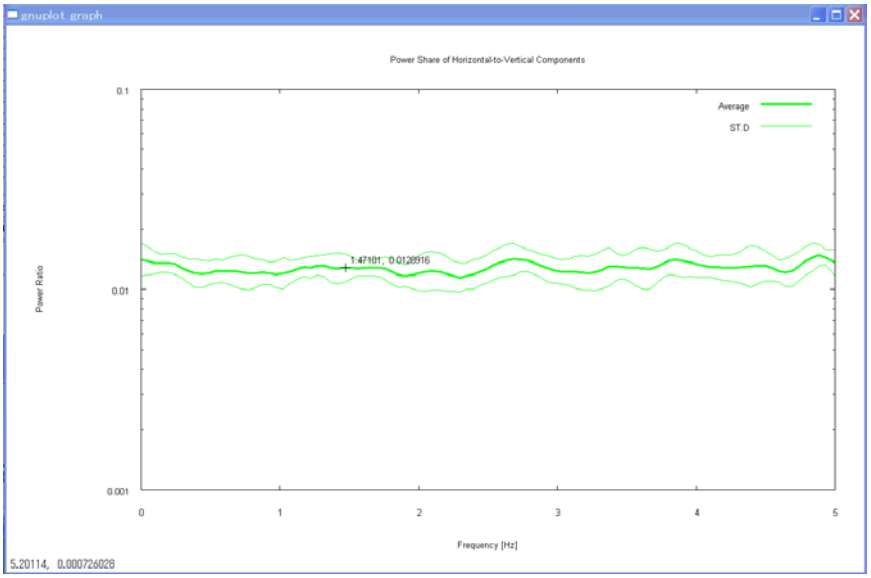
In creating the synthetic data we set the R/V spectrum at 0.1, irrespective of the frequency (A6.1). The analysis results returned comparable values in a frequency band centered on 1-2 Hz.

4) Pressing the return key again takes you to a plot of g_R, the power shares of Rayleigh waves in horizontal motion (g_R is defined as the fraction of the power of Rayleigh waves, with the total power of horizontal motion taken as unity. Defining g_L, the power share of Love waves, in a similar way, we have $g_R+g_L=1$).

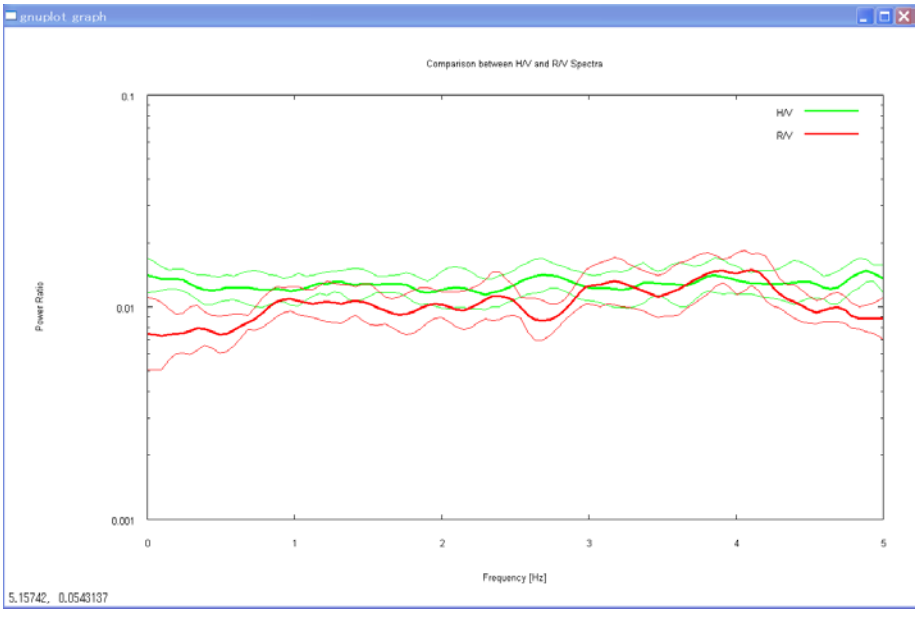


I set $g_R=0.8$ in the synthetic data (A6.1). The analysis results returned comparable values in a frequency band centered on 1-2 Hz.

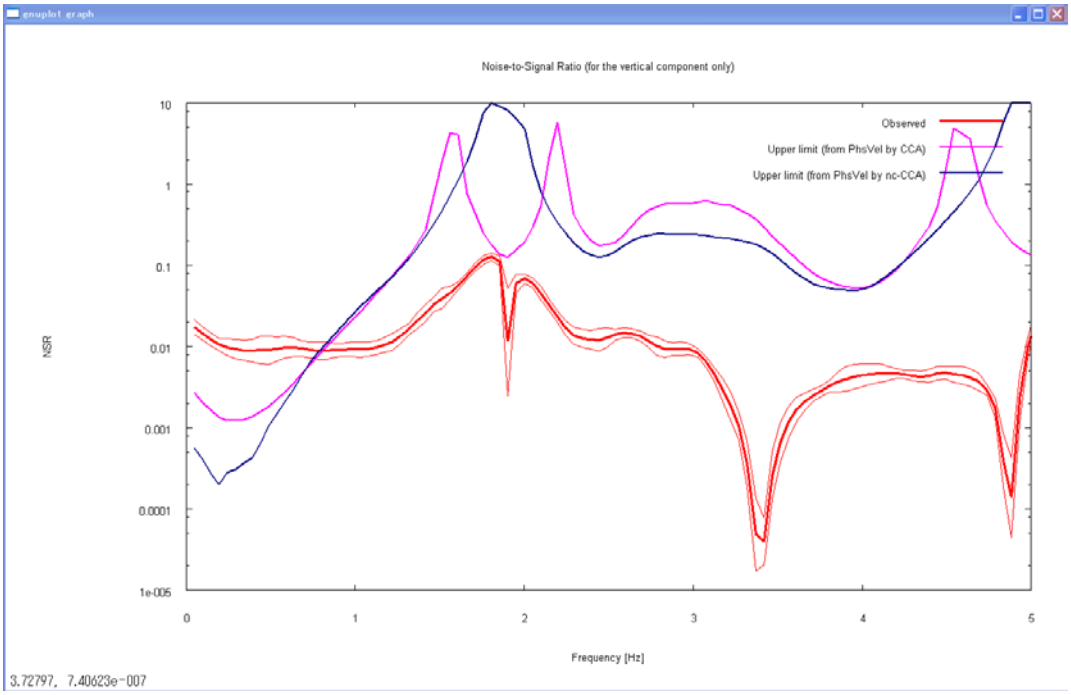
5) Pressing the return key again plots the H/V spectrum. According to calculations, the power ratio between the horizontal and vertical components should be equal to 0.0125. The analysis results do in fact return similar values.



6) Pressing the return key next time brings you to a comparative plot of the R/V spectrum of Rayleigh waves and the H/V spectrum. In this plot, the R/V spectrum is denoted in terms of power-spectral ratios (not in terms of amplitude ratios or ellipticities). As I said above in 3), the R/V equals 0.1 in terms of the amplitude ratio, so it should be equal to its square, 0.01, in terms of the power ratio. The given R/V and H/V values are equal to 0.01 and 0.0125, respectively, a difference represented well in the analysis results given here.



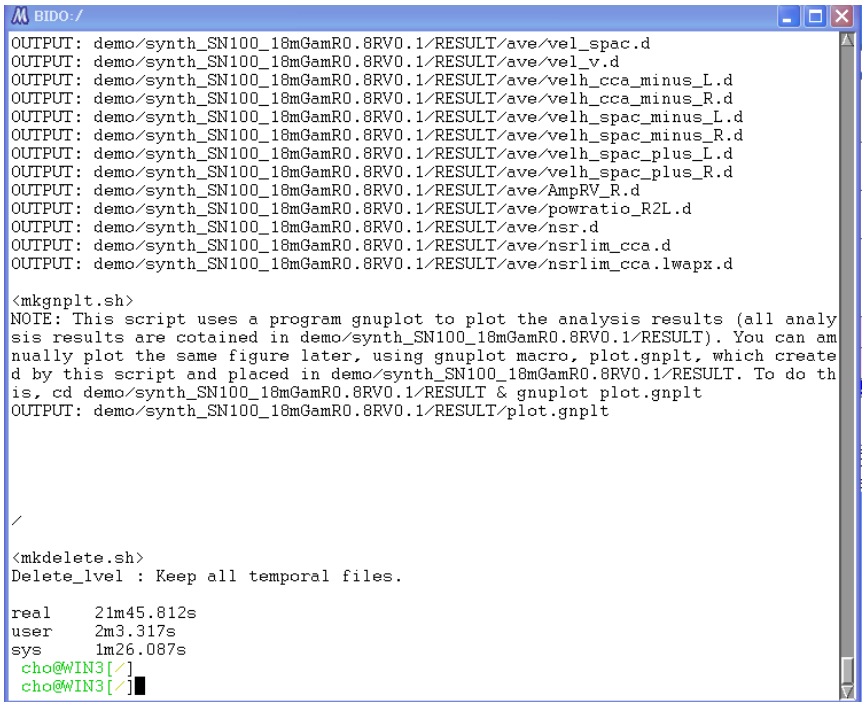
7) Pressing the return key again takes you to the final plot, that of the NS ratios (inverse of the SN ratios. Ratios of the power of incoherent noise to the power of signals) of vertical motion (red curves).



I set the SN ratio at 100 in the synthetic data (A6.1). This corresponds to an NS ratio of 0.01, because it is the inverse of the SN ratio. The analysis results returned comparable values in low-frequency ranges over 1 Hz.

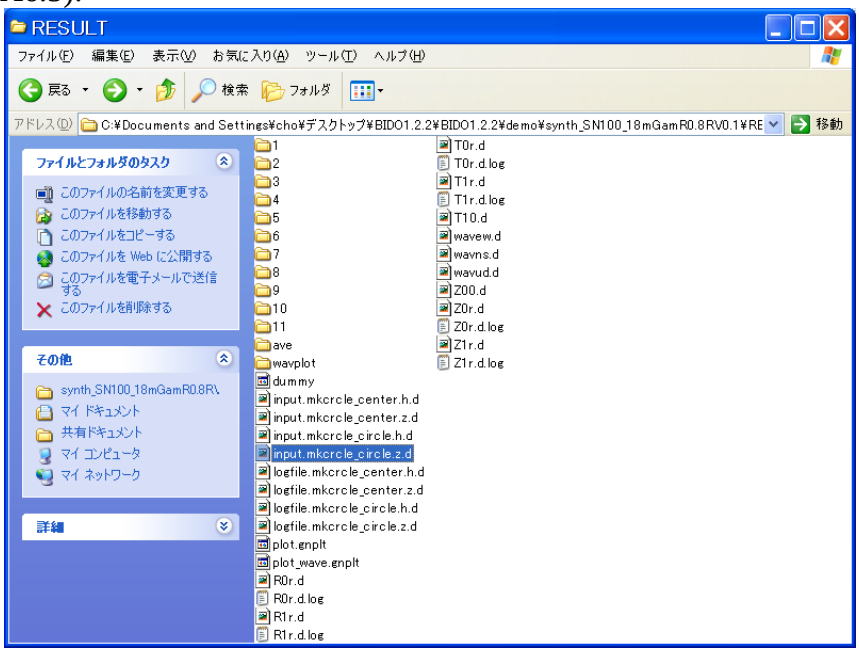
The pink and blue curves in the above figure show threshold NS ratios, calculated using phase velocity estimates from the CCA and nc-CCA methods, respectively. A threshold NS ratio is a reference value, used to evaluate the reliability of analysis results from the CCA method (not the nc-CCA method). When the observed NS ratios exceed the threshold NS ratios, the analysis results of the CCA method tend to be underestimates. In the above figure, the red comes above the pink (and above the blue) below about 0.7 Hz, so there is anxiety that the phase velocities inferred from the CCA method may be underestimates in that frequency band. A look at the plot in step 2) confirms that this is in fact the case.

This completes the plotting of the analysis results. When you press the return key, the plot screen disappears, and on the terminal you see:

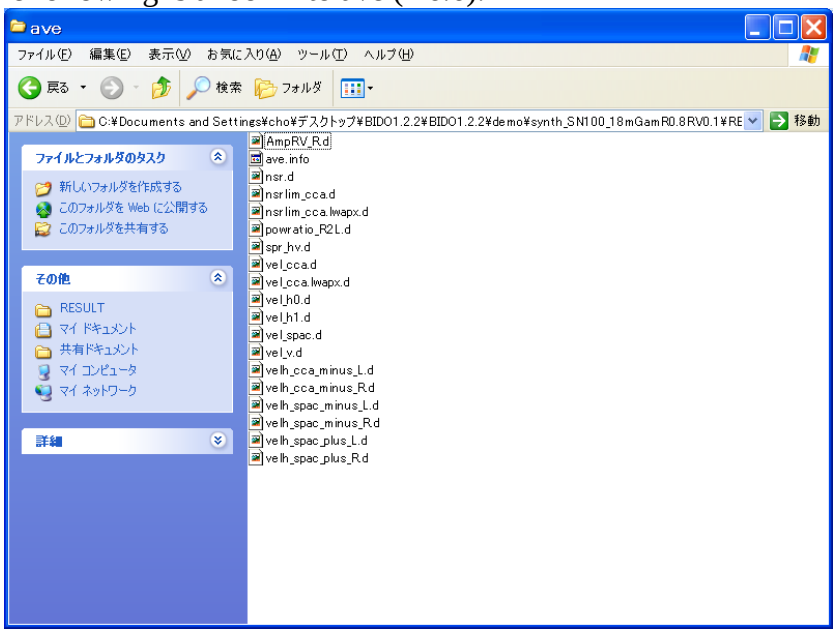


This shows that temporary files used in the analysis have been deleted at the end of the whole procedure. Typing "exit" at the command prompt or pressing C-Z (simultaneously pressing the control key and the z key) finishes the terminal screen.

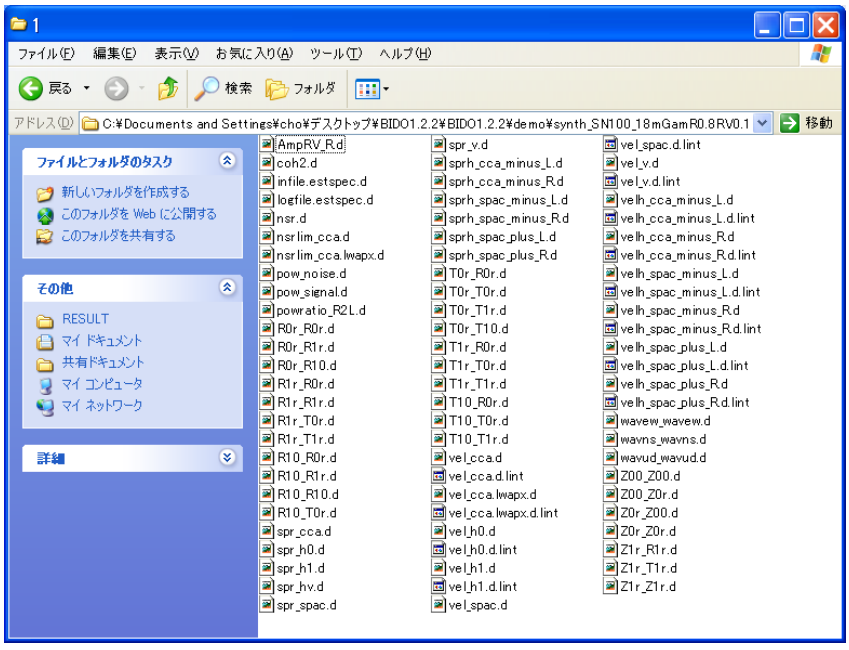
The whole analysis is over now. All analysis results are stored in a folder, named RESULT, which has been created beneath the data folder. The following is a look into RESULT (A6.5):



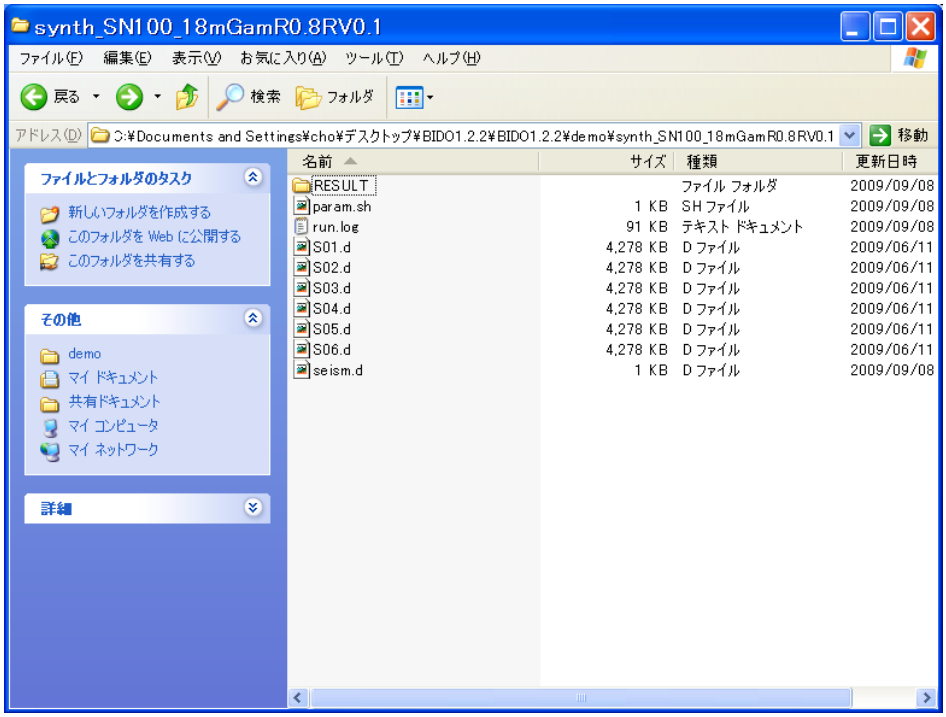
All data files used in the above plots are stored in the folder named ave (short for average). The following is a look into ave (A6.6).



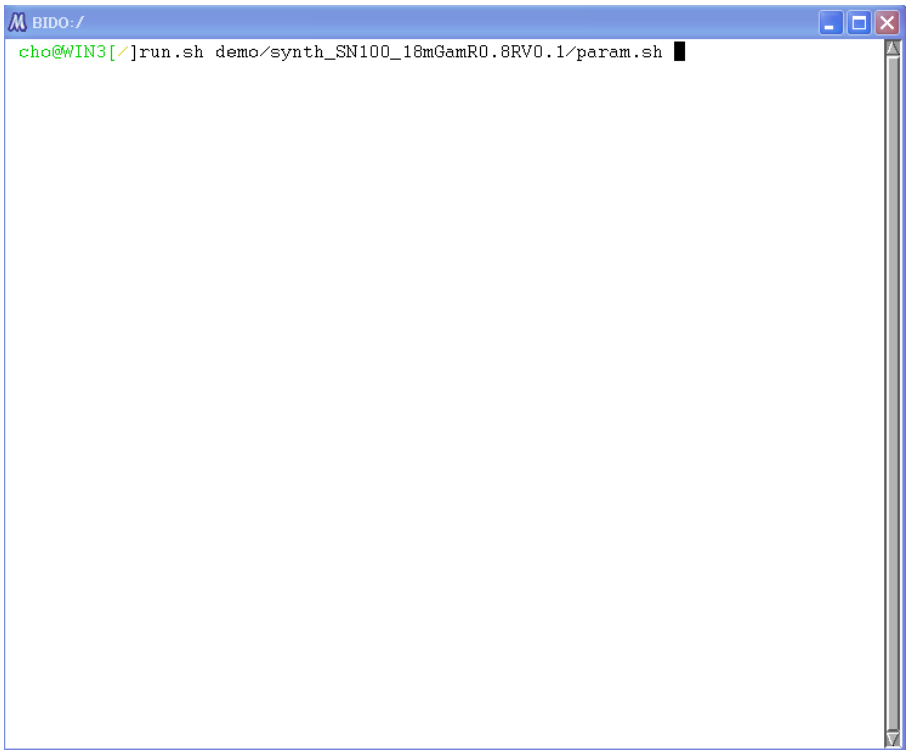
All files in the ave folder are statistical processings of individual analysis results that are stored in folders with numerical names such as 1, 2, etc. The numerically named folders contain not just the analysis results intended for statistical processing and storage in ave, but also a variety of analysis results (such as spectral densities and spectral ratios) that are normally not referred to unless you are really into in-depth analysis. For example, the folder named 1 contains the following analysis results (A6.7):



Coming back to the top of the data folder, you will find that an execution log, named run.log, and a parameter file, named param.sh, have been generated alongside the RESULT folder for the analysis results.



This parameter file can be conveniently used as an argument for run.sh when you rerun the analysis with parameters only slightly changed. See what follows:



You may be concerned about having to type the long path name or mistyping it. Feel at ease, though, because there is this typing-aid feature called auto-complete. Auto-complete allows you to type alphabetic keys only halfway down a word and press the TAB key so that the rest of the spelling turns up automatically. For example, if you press

run.sh d [TAB]

it automatically turns into

run.sh demo/

auto-complete is valid for the rest of the text. So, if you execute run.sh with the parameter file as an argument, the dialogue appearing at the start of the program asks you about the parameters of the last execution by default (when the parameters are not given explicitly like this, default_param.sh beneath the script folder is automatically referred to). Note that, when you rerun the analysis, all folders with numerical names such as 1, 2, etc. and an ave folder, beneath RESULT, are automatically deleted. If you do not wish those folders to be deleted, you have to move them somewhere else or to rename them before you rerun.

Finally, in the present example we have obtained analysis results for the (1) phase velocities of Love waves, (2) phase velocities of Rayleigh waves, (3) R/V spectra of Rayleigh waves, (4) power shares of Rayleigh waves in horizontal motion and (5) NS ratios of vertical motion, because we used three-component array data with a central sensor. In general, however, the analysis output (and graphic output) depend on the array geometry (whether it includes a central sensor) and the number of components in the waveform (vertical motion only, horizontal motion only or all three components). Take a try by editing the seism.d file to set #COMP at 1, 2 and 3 and comment out the line about the central station. In seism.d, appending # at the head of a line comments that line out, except for the lines with #DT and #COMP. Rerunning the analysis, you will obtain results for (2) and (5) alone for vertical-motion array data (#COMP 1) with a central station, and (2) alone for vertical-motion array data without a central station (# at the head of the third line). You will obtain (1) and (2) for horizontal-motion array data (#COMP 2) with or without a central station, and (1), (2) and (3) for three-component array data (#COMP 3) without a central station (# at the head of the third line).

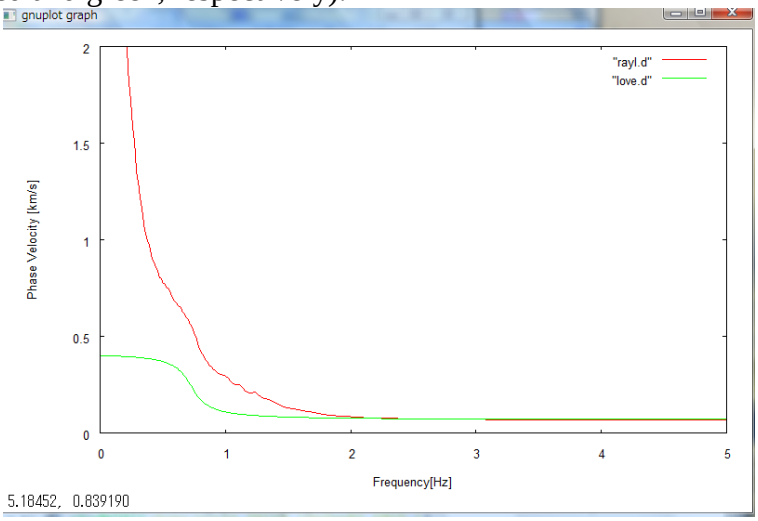
A6.1 Details of the Synthetic Data

The synthetic data were generated under the assumption that the field of microtremors satisfied the following conditions.

1) Chacateristics of the microtremor wavefield (signals)
- The field of microtremors is dominated by surface waves (Rayleigh and Love waves).
- The waves arrive as plane waves in the array.
- The wavefield is stationary in both time and space.
- The Rayleigh and Love waves are mutually uncorrelated.

2) Phase velocities of Rayleigh and Love waves

Phase velocities as illustrated in the figure below were prescribed to the Rayleigh and Love waves (red and green, respectively).



3) Arrival directions and intensities of Rayleigh and Love waves

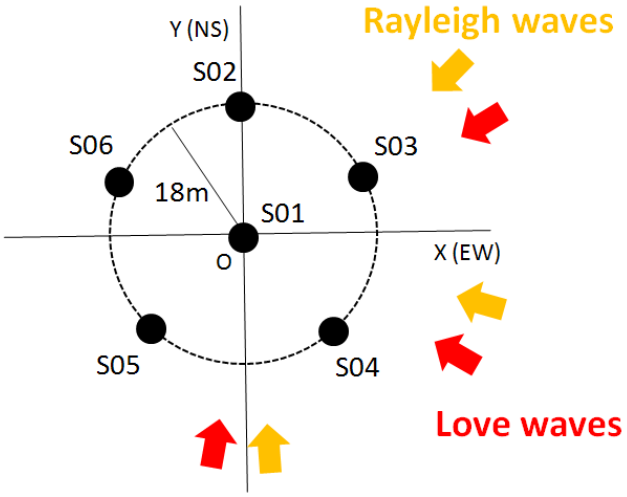
Rayleigh and Love waves were made to arrive as plane waves in the array as illustrated in the figure below. We assumed that both Rayleigh and Love waves arrived from all directions with equal intensities, but that Rayleigh and Love waves had different intensities. More specifically, we set the Rayleigh-to-Love power ratio at 4:1. The horizontal-to-vertical amplitude ratio of the Rayleigh waves was set at 1:10. The ratios were fixed at these values for all frequencies.

4) Noise

On top of the Rayleigh- and Love-wave signals we added incoherent noise with the following properties.

- Records of noise at different stations are mutually uncorrelated.
- Noise is stationary in time.
- Noise is uncorrelated with the signals.

We calculated the noise intensity corresponding to an SN ratio of 100 for all UD, NS and EW components, and added noise on top of the signals composed of Rayleigh and Love waves.



A6.2 The Dialogue that Appears on Activating the Program

The Dialogue that appears on activating the program asks about the following parameters. Enter appropriate values according to instructions written in **red.**

Data directory name [demo/synth_SN100_18mGamR0.8RV0.1] >← Enter the name of the folder containing data files. Separate the folder path with slashes (/).

* Execution parameters
Automatically select data portions to be used? (y/n=1/0)[1]←Setting this option to 1 means that portions of the data to be used in the estimation of spectral densities are selected automatically from the waveform data and that a segment file (A6.4) is generated. Create a segment file on your own and set this option to 0 if you wish to make your own selection of the portions of the waveform to be used in the analysis.
Calculate phase velocities and other properties of surface waves? (y/n=1/0)[1] ← Set this option to 1 to execute the data processing.
Delete temporary data files? (y/n=1/0)[0] ← Setting this option to 1 ensures that all intermediary data, generated on the way, are deleted when the data processing is over, except for the final analysis results and a small number of input files and log files. Setting this to 0 allows all intermediary data to be preserved.
Plot analysis results? (y/n=1/0)[1] ← Set this option to 1 to plot the analysis results.

* Basic parameters
Take a look at Estimating Spectral Densities (Section 5) and A Knack for Setting Parameters (A6.9) before proceeding to enter the following parameters.
Duration of data segments for the evaluation of spectra [s] [10.24] ← The segment duration [seconds] used in the estimation of spectral densities (segment averaging method). When the number of data points, calculated by (segment duration) / (sampling time interval), is not equal to a power of two, it is automatically zero-padded to a power of two during FFT (for the sake of efficiency, it is recommendable to set values so that it equals a power of two).
Number of data segments over which averages are taken. Enter 0 or a very large number if you wish to use all segments simultaneously [10] ← The number of segments (integer) used in the estimation of spectral densities (segment averaging method). Entering a large integer, which exceeds the total number of segments catalogued in the segment file, or zero means "segment averaging for a single segment (virtually no segment averaging)."
Band width of the Parzen spectral window [Hz] [0.3] ← The width of the spectral (Parzen) window [Hz] used in the estimation of spectral densities (smoothing method). Entering zero means "a zero window width (virtually no use of a spectral window)."

* Data file and array geometry
A preexisting seismfile demo/synth_SN100_18mGamR0.8RV0.1/seism.d (for the data file names and the array geometry) has been detected. ← Prescribe the array geometry, waveform components and the sampling time interval in \(data folder)\seism.d. When that file is not found, interactive questions generate one automatically.
----------citation begins here----------

#COMP 3 (1 ud/ 2 ns & ew / 3 three components) ← Enter the waveform components (type 1, 2 or 3 following a space or a tab after #COMP).
#DT 0.01 ← Enter the time interval of waveform sampling (type a figure following a space or a tab after #DT).
0.000000 0.000000 S01.d 1 ← From left to right, the x coordinate [km], y coordinate [km], data file name, and integer 1 or 0 (1 if center).
0.000001 0.018000 S02.d 0
0.017119 0.005562 S03.d 0
0.010581 -0.014562 S04.d 0
-0.010579 -0.014563 S05.d 0
-0.017119 0.005564 S06.d 0

Notes
* Appending "#" at the head of a line comments that line out (except for #COMP and #DT). There is no rule on the order of arrangement of these data.
* When there are only two sensors constituting the array, think of either one of them as being at the center and the other as lying around the circle, and accordingly set integers of 1 and 0 to the right of the data file names.
* When there is only one sensor constituting the array, the integer to the right of the data file name can be either 0 or 1 (the H/V spectrum alone is calculated, and there will be no array processing).
* When the records have three components (#COMP 3), the H/V spectrum is calculated using the file indicated in the top data line.

A6.3 The seismfile

Description

A file prescribing the array geometry, waveform components and the sampling time interval.

File Name

\(data folder)\RESULT\seism.d

Format

#COMP 3 (1 ud/ 2 ns & ew / 3 three components) ßEnter the waveform components (type 1, 2 or 3 following a space or a tab after #COMP).
#DT 0.01 ßEnter the time interval of waveform sampling (type a figure following a space or a tab after #DT).
0.000000 0.000000 S01.d 1 ßFrom left to right, the x coordinate [km], y coordinate [km], data file name, and integer 1 or 0 (1 if center).
0.000001 0.018000 S02.d 0
0.017119 0.005562 S03.d 0
0.010581 -0.014562 S04.d 0
-0.010579 -0.014563 S05.d 0
-0.017119 0.005564 S06.d 0

Notes
* Appending "#" at the head of a line comments that line out (except for #COMP and #DT). There is no rule on the order of arrangement of these data.

* When there are only two sensors constituting the array, think of either one of them as being at the center and the other as lying around the circle, and accordingly set integers of 1 and 0 to the right of the data file names.
* When there is only one sensor constituting the array, the integer to the right of the data file name can be either 0 or 1 (the H/V spectrum alone is calculated, and there will be no array processing).
* When the records have three components (#COMP 3), the H/V spectrum is calculated using the file indicated in the top data line.

A6.4 The segment File

Description

A file indicating which segment portions should be extracted from the waveform data.

File Name

The file is named differently for different components contained in the waveform data.
- Vertical motion alone is contained → \(data folder)\RESULT\segment_z.d
- Horizontal motion alone is contained → \(data folder)\RESULT\segment_h.d
- All three components are contained → \(data folder)\RESULT\segment_3c.d

Format

115 ←Total number of segments
10.24 ←Segment duration
0.01 ←Time interval of waveform sampling
0. ←Start time of the first segment
5.12 ←Start time of the second segment
10.24 ←Start time of the third segment
...
Numbers should be laid out like this, one in each line. There is no particular format for the numbers (except that an integer alone is allowed in the first line).

A6.5 Analysis Results (folder RESULT)

1, 2, ..., *n*, ... Folder containing analysis results relevant to the *n*th segment cluster
ave Folder containing data files used in gnuplot plots (files with statistics of the analysis results in folders 1, 2, ...)
R0r.d Zeroth-order Fourier coefficients for the radial component of measurement data around the circle
R0r.d.log Execution log of the code that generates R0r.d
R10.d First-order Fourier coefficients for the radial component of measurement data at the center
R1r.d First-order Fourier coefficients for the radial component of measurement data around the circle
R1r.d.log Execution log of the code that generates R1r.d
T0r.d Zeroth-order Fourier coefficients for the tangential component of measurement data around the circle
T0r.d.log Execution log of the code that generates T0r.d
T10.d First-order Fourier coefficients for the tangential component of measurement data at the center

T1r.d First-order Fourier coefficients for the tangential component of measurement data around the circle
T1r.d.log Execution log of the code that generates T1r.d
Z00.d Zeroth-order Fourier coefficients for the UD component of measurement data at the center
Z0r.d Zeroth-order Fourier coefficients for the UD component of measurement data around the circle
Z0r.d.log Execution log of the code that generates Z0r.d
Z1r.d First-order Fourier coefficients for the UD component of measurement data around the circle
Z1r.d.log Execution log of the code that generates Z1r.d
wavud.d Copy of the data file indicated first in the seism file. Used to estimate the H/V spectrum.
wavns.d Copy of the data file indicated first in the seism file. Used to estimate the H/V spectrum.
wavew.d Copy of the data file indicated first in the seism file. Used to estimate the H/V spectrum.
dummy Used in gnuplot plots. A dummy data file that contains nothing.
input.mkcrcle_center.h.d Input data for the calculation of Fourier coefficients at the center (horizontal components)
input.mkcrcle_center.z.d Input data for the calculation of Fourier coefficients at the center (UD component)
input.mkcrcle_circle.h.d Input data for the calculation of Fourier coefficients around the circle (horizontal components)
input.mkcrcle_circle.z.d Input data for the calculation of Fourier coefficients around the circle (UD component)
logfile.mkcrcle_center.h.d Execution log of the code that calculates Fourier coefficients at the center (horizontal components)
logfile.mkcrcle_center.z.d Execution log of the code that calculates Fourier coefficients at the center (UD component)
logfile.mkcrcle_circle.h.d Execution log of the code that calculates Fourier coefficients around the circle (horizontal components)
logfile.mkcrcle_circle.z.d Execution log of the code that calculates Fourier coefficients around the circle (UD component)
plot.gnplt Macro to activate gnuplot
segment_3c.d Segment file (A6.4) describing which segments were extracted from the measurement data
segment_3c.d.histogram Histogram of RMSs used in the automatic selection of the segments

A6.6 Analysis Results (folder ave)

AmpRV_R.d Horizontal-to-vertical amplitude ratios of Rayleigh waves (averaged logarithmically)
ave.info Rough explanation of how the analysis results were averaged
nsr.d NS ratios of vertical motion (averaged logarithmically)
nsrlim_cca.d Threshold NS ratios calculated with phase velocity estimates of the CCA method (5% tolerance for relative error) (averaged logarithmically)

nsrlim_cca.lwapx.d Threshold NS ratios calculated with phase velocity estimates of the nc-CCA method (5% tolerance for relative error) (averaged logarithmically)
powratio_R2L.d Power shares of Rayleigh waves in the total power of horizontal motion (averaged logarithmically)
spr_hv.d Ratios of the power of horizontal motion (sum of the two horizontal-component powers) to the power of vertical motion (averaged logarithmically)
vel_cca.d Phase velocities of Rayleigh waves according to the CCA method
vel_cca.lwapx.d Phase velocities of Rayleigh waves according to the nc-CCA method
vel_h0.d Phase velocities of Rayleigh waves according to the H0 method
vel_h1.d Phase velocities of Rayleigh waves according to the H1 method
vel_spac.d Phase velocities of Rayleigh waves according to the SPAC method (spatial autocorrelation coefficients according to an original definition)
vel_v.d Phase velocities of Rayleigh waves according to the V method
velh_cca_minus_L.d Phase velocities of Love waves according to the CCA-L method
velh_cca_minus_R.d Phase velocities of Rayleigh waves according to the CCA-R method
velh_spac_minus_L.d Phase velocities of Love waves according to the SPAC-L method
velh_spac_minus_R.d Phase velocities of Rayleigh waves according to the SPAC-R method
velh_spac_plus_L.d Phase velocities of Love waves according to the SPAC+L method
velh_spac_plus_R.d Phase velocities of Rayleigh waves according to the SPAC+R method
* In the above data files, the frequency, mean of segment-specific analysis results and standard deviation are laid out in the first, second and third columns, respectively. In data files with the note (averaged logarithmically) in parentheses, the frequency, mean, mean minus a standard deviation and mean plus a standard deviation are laid out in the first, second, third and fourth columns, respectively.

A6.7 Analysis Results (folders with alphanumeric names)

infile.estspec.d Input data for the estimation of spectral densities
logfile.estspec.d Execution log of the code that estimates spectral densities

The following are spectral density estimates. For example, Z1r_R1r.d stands for the cross-spectral densities between the Z (UD)-component, first-order, circumferential data and the radial (R)-component, zeroth-order, circumferential data.
Z00_Z00.d Z00_Z0r.d Z0r_Z00.d Z0r_Z0r.d Z1r_R1r.d Z1r_T1r.d Z1r_Z1r.d
R0r_R0r.d R0r_R10.d R0r_R1r.d R10_R0r.d R10_R10.d R10_R1r.d R10_T0r.d
R1r_R0r.d R1r_R1r.d R1r_T0r.d R1r_T1r.d T0r_R0r.d T0r_T0r.d T0r_T10.d
T0r_T1r.d T10_R0r.d T10_T0r.d T10_T1r.d T1r_R0r.d T1r_T0r.d T1r_T1r.d

coh2.d Magnitude-squared coherences between the vertical-motion, zeroth-order, circumferential data and the central data
nsr.d NS ratios of vertical motion
nsrlim_cca.d Threshold NS ratios calculated with phase velocity estimates of the CCA method (5% tolerance for relative error)
nsrlim_cca.lwapx.d Threshold NS ratios calculated with phase velocity estimates of the nc-CCA method (5% tolerance for relative error)
pow_noise.d Power of noise at the center, vertical motion
pow_signal.d Power of signals at the center, vertical motion
AmpRV_R.d Horizontal-to-vertical amplitude ratios of Rayleigh waves
powratio_R2L.d Power shares of Rayleigh waves in the total power of horizontal motion

spr_cca.d Spectral ratios used to infer phase velocities of Rayleigh waves with the CCA method
spr_cca.lwapx.d Spectral ratios used to infer phase velocities of Rayleigh waves with the nc-CCA method
spr_h0.d Spectral ratios used to infer phase velocities of Rayleigh waves with the H0 method
spr_h1.d Spectral ratios used to infer phase velocities of Rayleigh waves with the H1 method
spr_hv.d Ratios of the power of horizontal motion (sum of the two horizontal-component powers) to the power of vertical motion
spr_spac.d Spectral ratios used to infer phase velocities of Rayleigh waves with the SPAC method (spatial autocorrelation coefficients according to an original definition)
spr_v.d Spectral ratios used to infer phase velocities of Rayleigh waves with the V method
sprh_cca_minus_L.d Spectral ratios used to infer phase velocities of Love waves with the CCA-L method
sprh_cca_minus_R.d Spectral ratios used to infer phase velocities of Rayleigh waves with the CCA-R method
sprh_spac_minus_L.d Spectral ratios used to infer phase velocities of Love waves with the SPAC-L method
sprh_spac_minus_R.d Spectral ratios used to infer phase velocities of Rayleigh waves with the SPAC-R method
sprh_spac_plus_L.d Spectral ratios used to infer phase velocities of Love waves with the SPAC+L method
sprh_spac_plus_R.d Spectral ratios used to infer phase velocities of Rayleigh waves with the SPAC+R method

vel_cca.d Phase velocities of Rayleigh waves according to the CCA method
vel_cca.lwapx.d Phase velocities of Rayleigh waves according to the nc-CCA method
vel_h0.d Phase velocities of Rayleigh waves according to the H0 method

vel_h1.d Phase velocities of Rayleigh waves according to the H1 method
vel_spac.d Phase velocities of Rayleigh waves according to the SPAC method (spatial autocorrelation coefficients according to an original definition)
vel_v.d Phase velocities of Rayleigh waves according to the V method
velh_cca_minus_L.d Phase velocities of Love waves according to the CCA-L method
velh_cca_minus_R.d Phase velocities of Rayleigh waves according to the CCA-R method
velh_spac_minus_L.d Phase velocities of Love waves according to the SPAC-L method
velh_spac_minus_R.d Phase velocities of Rayleigh waves according to the SPAC-R method
velh_spac_plus_L.d Phase velocities of Love waves according to the SPAC+L method
velh_spac_plus_R.d Phase velocities of Rayleigh waves according to the SPAC+R method

vel*.d.lint Data resampled for the statistical processing of phase velocities

A6.8 One Approach to Make the Most of the Method's Potential

The CCA method, one of the analysis methods adopted in our codes, allows one to analyze waves of very long wavelengths when the SN ratios (signal-to-noise power ratios) are sufficiently good. Presence of noise sources near the seismic array lowers SN ratios, but it is difficult to control microtremor noise sources in urban areas where humans live and industries are active. The only solution would be to use "miniature arrays" with very small sizes (if the array lies within the reach of arms, it should be easy to ensure a calm environment in its surroundings, at least while the measurement is active [about 15 minutes]). The pictures in the top page show how we realized large SN ratios by using an array with a radius of just 30 cm. This array realized SN ratios in excess of 10,000, which made it possible to analyze wavelengths larger than 500 times the array radius (Reference [6]).

A6.9 A Knack for Setting Parameters

One major feature of the analysis methods adopted in our codes lies in their ability to analyze long wavelengths. A knack for setting analysis parameters to make the most of this feature is to make the segment duration as small as possible within the tolerable range that depends on the frequency band of interest you wish to look at. This measure is effective to extracting a large number of segments from observed waveforms that are contaminated by non-stationary noise, and helps to make as narrow as possible the bandwidth of smoothing with a spectral window. Spectral windowing can cause biases, which becomes a particularly critical problem in the analysis of long wavelength ranges **(**Reference [3]**).**

7. Warnings / Download

<Warnings>
- This program is distributed free with source codes.
- Do not give this program to a third party (a third party should download it from this Web site).
- We assume no responsibility for any problems arising from the use of this program.
- Consult with us in advance if you wish to use this program for commercial purposes.
- Be sure to make appropriate citations when you publish research achievements benefiting from this program.

If you agree to the above provisions, register your e-mail address (to be used for possible information of bugs and for statistical purposes only) via a web site **http://staff.aist.go.jp/ikuo-chou/bidodl_en.html** and click OK to proceed to the download screen.

<Download>
This archive is common to both Windows and Linux.

BIDO1.2.tgz Source files + executable files + synthetic demo data (19 MBytes)

demo_obsz.tgz Sample data for vertical motion (worth two arrays at two sites—AIST and KSKB) (42 MBytes)

demo_obs3c.tgz Sample data for three components (worth six arrays at two sites—KSKB and KIBA) (74 MBytes)

A7 Citation

This program remains unpublished in itself, so we would be pleased if you could just state in your publication that you downloaded from the web site http://staff.aist.go.jp/ikuo-chou the analysis software BIDO that is based on our microtremor analysis theory [2, 5]. Citations regarding specific aspects should be: [2] for the general theory, [6] for miniature arrays, [3] for NS ratios of vertical motion, [2] for horizontal-to-vertical amplitude ratios of Rayleigh waves, [1, 3] for the CCA method, [4] for the H0, H1, V and nc-CCA methods, and [5] for the CCA-L (R), SPAC-L (R) and SPAC+L (R) methods and for Rayleigh-to-Love power ratios.

[1] Cho, I., T. Tada, and Y. Shinozaki, 2004, A new method to determine phase velocities of Rayleigh waves from microseisms, *Geophysics*, **69**, 1535-1551.

[2] Cho, I., T. Tada, and Y. Shinozaki, 2006, A generic formulation for microtremor exploration methods using three-component records from a circular array, *Geophys. J. Int.*, **165**, 236-258.

[3] Cho, I., T. Tada, and Y. Shinozaki, 2006, Centerless circular array method: Inferring phase velocities of Rayleigh waves in broad wavelength ranges using microtremor records, *J. Geophys. Res.*, **111**, B09315, doi:10.1029/2005JB004235.

[4] Tada, T., I. Cho, and Y. Shinozaki, 2007, Beyond the SPAC method: Exploiting the wealth of circular-array methods for microtremor exploration, *Bull. Seism. Soc. Am.*, **97**, 2080-2095, doi:10.1785/0120070058.

[5] Tada, T., I. Cho, and Y. Shinozaki, 2009, New circular-array microtremor techniques to infer Love-wave phase velocities, *Bull. Seism. Soc. Am.*, **99**, 2912-2926, doi:10.1785/0120090014.

[6] Cho, I., T. Tada, and Y. Shinozaki, 2008, A new method of microtremor exploration using miniature seismic arrays: Quick estimation of average shear velocities of the shallow soil (in Japanese with English abstract), *Butsuri-Tansa*, **61**, 457-468.

Appendix Execution log (run.log)

```
OS windows
paramfile:./param.sh
export datadir=demo/synth_SN100_18mGamR0.8RV0.1
export segment_duration=10.24
export smoothband=0.3
export nseg_segave=10
export idmksegment=1
export idcalc=1
export idgnplt=1
export delete_level=0
<SETPAR>
Three-component waveforms will be used [according to demo/synth_SN100_18mGamR0.8RV0.1/seism.d].
(Analysis includes the estimation of horizontal-to-vertical spectral ratios.)
<MKSEGMENT>
Output demo/synth_SN100_18mGamR0.8RV0.1/RESULT/segment_3c.d
[Segments with RMS values between 1.000000 and 1.200000 are selected]
NOTE: 115 segments have been selected.

<mkgnplt_wave.sh>
NOTE: This script uses a program gnuplot to plot the waveforms. You can manually plot the same figures later, using a gnuplot
macro, plot_wave.gnplt, which was created by this script and is found in demo/synth_SN100_18mGamR0.8RV0.1/RESULT. To do
this, type cd demo/synth_SN100_18mGamR0.8RV0.1/RESULT & gnuplot plot_wave.gnplt
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/plot_wave.gnplt
/
<MKCIRCLE>
seismfile: demo/synth_SN100_18mGamR0.8RV0.1/seism.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/Z0r.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/Z1r.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/Z00.d
<MKCIRCLE_H>
seismfile: demo/synth_SN100_18mGamR0.8RV0.1/seism.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/R0r.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/R1r.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/T0r.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/T1r.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/R10.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/T10.d
<MKWAV3C>
seismfile: demo/synth_SN100_18mGamR0.8RV0.1/seism.d
demo/synth_SN100_18mGamR0.8RV0.1/S01.d used to estimate horizontal-to-vertical spectral ratios
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/wavud.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/wavew.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/wavns.d

* Preexisting directori(es) detected (created in a previous calculation?):
demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1 demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave
NOTE: These directories are removed to proceed with the current analysis.


### Step 1 (Total 11) ###
<SETPAR>
Three-component waveforms will be used [according to demo/synth_SN100_18mGamR0.8RV0.1/seism.d].
(Analysis includes the estimation of horizontal-to-vertical spectral ratios.)
<ESTSPEC>
NOICE:idrimag=1, read columns 1-3
[column 1: time; 2: real part; 3: imaginary part]
OUTPUT DIR: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1
 No. of Seismograms: 9
DATAFILE  1) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/R10.d
DATAFILE  2) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/R0r.d
DATAFILE  3) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/R1r.d
DATAFILE  4) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/T10.d
DATAFILE  5) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/T0r.d
DATAFILE  6) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/T1r.d
DATAFILE  7) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/Z00.d
DATAFILE  8) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/Z0r.d
DATAFILE  9) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/Z1r.d
```

```
Cross-spectral density calculations (1:yes; 2: no)
  | 1 2 3 4 5 6 7 8 9
--------------------------------
  1| 1 1 1 0 1 0 0 0 0
  2| 1 1 1 0 0 0 0 0 0
  3| 0 1 1 0 1 1 0 0 0
  4| 0 1 0 0 1 1 0 0 0
  5| 0 1 0 1 1 1 0 0 0
  6| 0 1 0 0 1 1 0 0 0
  7| 0 0 0 0 0 0 1 1 0
  8| 0 0 0 0 0 0 1 1 0
  9| 0 0 1 0 0 1 0 0 1

No. of segments:    10
Data segment duration: 10.240 s
dt          :  0.010 s
Total duration (zero-padded):   20.480s (   2048 pts)
Ratio of data length to total length (zero-padded):  0.500
(Power reduction is corrected based on this value)
Frequency interval:0.488E-01 Hz
Nyquist frequency : 50.0     Hz
Use data window  [Hanning window with taper rate: 0.500 (  513 pts)]
Ratio of tapered power to raw power:  0.50
(Power reduction is corrected based on this value)
Use spectral window [Parzen window with band width:  0.30000 Hz (Total   13 pts)]
Resulting degree of freedom:    65
Standard error:     0.176777 (the effects of cos taper & zero-padding considered)
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/R10_R10.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/R10_R0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/R10_R1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/R10_T0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/R0r_R10.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/R0r_R0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/R0r_R1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/R1r_R0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/R1r_R1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/R1r_T0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/R1r_T1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/T10_R0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/T10_T0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/T10_T1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/T0r_R0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/T0r_T10.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/T0r_T0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/T0r_T1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/T1r_R0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/T1r_T0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/T1r_T1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/Z00_Z00.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/Z00_Z0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/Z0r_Z00.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/Z0r_Z0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/Z1r_R1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/Z1r_T1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/Z1r_Z1r.d
<ESTSPEC>
NOICE:idrimag=    0 read only columns 1 and 2
No other columns are read
OUTPUT DIR: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1
 No. of Seismograms: 3
DATAFILE  1) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/wavud.d
DATAFILE  2) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/wavew.d
DATAFILE  3) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/wavns.d

Cross-spectral density calculations (1:yes; 2: no)
  | 1 2 3
-------------
  1| 1 0 0
  2| 0 1 0
  3| 0 0 1
```

No. of segments: 10
Data segment duration: 10.240 s
dt : 0.010 s
Total duration (zero-padded): 20.480s (2048 pts)
Ratio of data length to total length (zero-padded): 0.500
(Power reduction is corrected based on this value)
Frequency interval:0.488E-01 Hz
Nyquist frequency : 50.0 Hz
Use data window [Hanning window with taper rate: 0.500 (513 pts)]
Ratio of tapered power to raw power: 0.50
(Power reduction is corrected based on this value)
Use spectral window [Parzen window with band width: 0.30000 Hz (Total 13 pts)]
Resulting degree of freedom: 65
Standard error: 0.176777 (the effects of cos taper & zero-padding considered)
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/wavud_wavud.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/wavew_wavew.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/wavns_wavns.d
<SPECRATIO>
OUTPUT (for CCA) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/spr_cca.d
OUTPUT (for SPAC) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/spr_spac.d
OUTPUT (for H0) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/spr_h0.d
OUTPUT (for MSC) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/coh2.d
Array radius: 0.018000 [from demo/synth_SN100_18mGamR0.8RV0.1/RESULT/Z1r.d.log]
OUTPUT (for NSR) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/nsr.d
OUTPUT (for signal power) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/pow_signal.d
OUTPUT (for noise power) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/pow_noise.d
OUTPUT (for H1) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/spr_h1.d
OUTPUT (for V) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/spr_v.d
<SPECRATIO for horizontal components>
OUTPUT (for CCA-L) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/sprh_cca_minus_L.d
OUTPUT (for CCA-R) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/sprh_cca_minus_R.d
OUTPUT (for R/V) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/AmpRV_R.d
OUTPUT (for SPAC-L) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/sprh_spac_minus_L.d
OUTPUT (for SPAC+L) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/sprh_spac_plus_L.d
OUTPUT (for SPAC-R) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/sprh_spac_minus_R.d
OUTPUT (for SPAC+R) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/sprh_spac_plus_R.d
OUTPUT (for R/(R+L)) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/powratio_R2L.d
<SPECRATIO HV>
OUTPUT (for HV spectral ratio) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/spr_hv.d
<SPEC2PV>
Array radius: 0.018000 [from demo/synth_SN100_18mGamR0.8RV0.1/RESULT/Z1r.d.log]
OUTPUT (for CCA) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/vel_cca.d
OUTPUT (for SPAC) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/vel_spac.d
OUTPUT (for H0) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/vel_h0.d
OUTPUT (for H1) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/vel_h1.d
OUTPUT (for V) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/vel_v.d
OUTPUT (for nc-CCA) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/vel_cca.lwapx.d
OUTPUT (for NSR limit by CCA) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/nsrlim_cca.d
OUTPUT (for NSR limit by nc-CCA) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/nsrlim_cca.lwapx.d
<SPEC2PV for horizontal components>
Array radius: 0.018000 [from demo/synth_SN100_18mGamR0.8RV0.1/RESULT/R0r.d.log]
OUTPUT (for CCA-L) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/velh_cca_minus_L.d
OUTPUT (for CCA-R) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/velh_cca_minus_R.d
OUTPUT (for SPAC-L) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/velh_spac_minus_L.d
OUTPUT (for SPAC+L) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/velh_spac_plus_L.d
OUTPUT (for SPAC-R) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/velh_spac_minus_R.d
OUTPUT (for SPAC+R) demo/synth_SN100_18mGamR0.8RV0.1/RESULT/1/velh_spac_plus_R.d

.....

Averaging over 11 segment clusters...
For the sake of robustness, the maximum and minimum values (plus non-numbers (ex. NaN)) are eliminated from the averaging process when the number of segment clusters exceeds NROBUST4AVERAGE_INC (defined in the file PARAM.h). Read the file demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/ave.info for details.

OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/vel_cca.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/vel_cca.lwapx.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/vel_h0.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/vel_h1.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/vel_spac.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/vel_v.d

OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/velh_cca_minus_L.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/velh_cca_minus_R.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/velh_spac_minus_L.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/velh_spac_minus_R.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/velh_spac_plus_L.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/velh_spac_plus_R.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/AmpRV_R.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/powratio_R2L.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/nsr.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/nsrlim_cca.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/nsrlim_cca.lwapx.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/ave/spr_hv.d

<mkgnplt.sh>
NOTE: This script uses a program gnuplot to plot the analysis results (all analysis results are contained in demo/synth_SN100_18mGamR0.8RV0.1/RESULT). You can manually plot the same figures later, using a gnuplot macro, plot.gnplt, which was created by this script and is found in demo/synth_SN100_18mGamR0.8RV0.1/RESULT. To do this, type cd demo/synth_SN100_18mGamR0.8RV0.1/RESULT & gnuplot plot.gnplt
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1/RESULT/plot.gnplt

<mkdelete.sh>
All temporary files preserved.